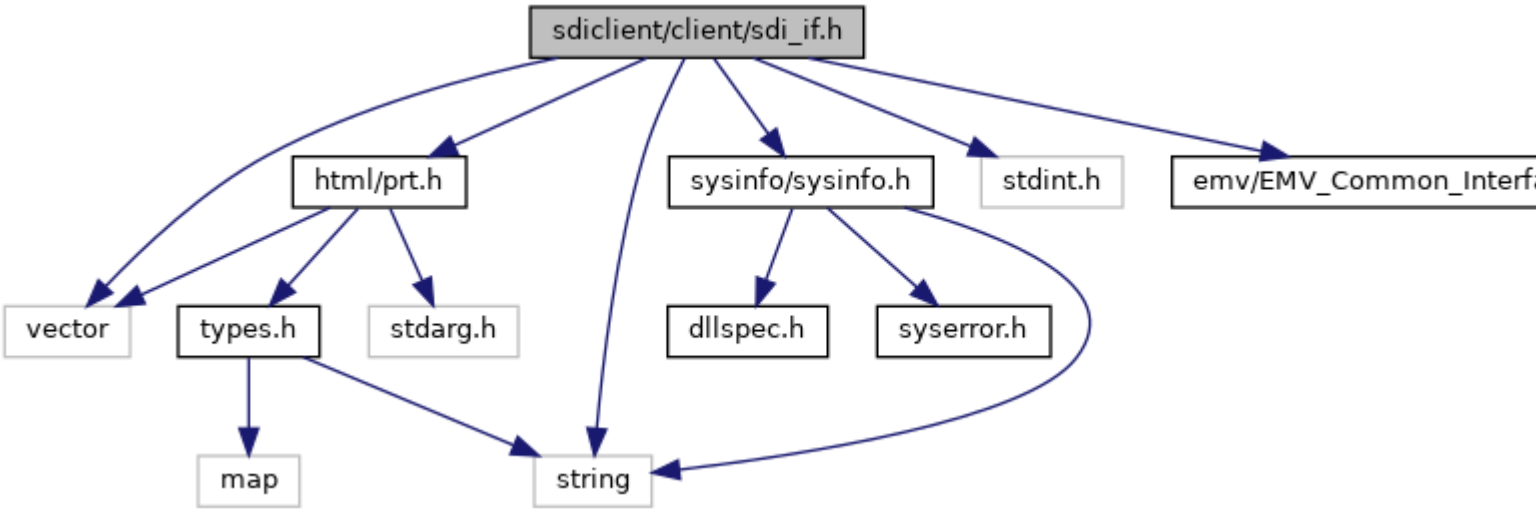
sdi_if.h File Reference

```
#include <vector>

#include <string>

#include <stdint.h>

#include <emv/EMV_Common_Interface.h>

#include <sysinfo/sysinfo.h>

#include <html/prt.h>
```

Include dependency graph for sdi_if.h:



Go to the source code of this file.

Data Structures

struct	MatchingRecord
class	SdiBase
struct	SdiBase::PluginResult
class	SdiCmd
	Composition for TLV based SDI commands. More...

class SDI

class CardDetection

Interface for SDI Card Detection Interface, command class 23. More...

class PED

class SdiCrypt

struct SdiCrypt::Placeholder

class ManualEntry

Interface for SDI command MSR Card Data Entry (21-02) More...

class Dialog

Namespaces

libsdi

Macros

#define MSR_CLIENT_ERROR_OFFSET 100

if a client error happens, msr functions will return (enum SDICLIENT_ERROR - MSR_CLIENT_ERROR_OFFSET) More...

#define VALIDATION_CHECK_OPTION_RETURN_ALL_MATCHING_RANGES 0x01

Enumerations

SDI_SW12 {

SDI_SW12_NONE = 0,
SDI_SW12_SUCCESS = 0x9000,
SDI_SW12_TAG_ERROR = 0x6200,
SDI_SW12_TAG_LENGTH_ERROR = 0x6300,

SDI_SW12_EXEC_ERROR = 0x6400,
SDI_SW12_CANCELED_BY_USER = 0x6405,
SDI_SW12_BUSY = 0x640A,
SDI_SW12_TIMEOUT_PIN_ENTRY = 0x640C,

SDI_SW12_TIMEOUT_NO_MSR_DATA = 0x64F6,
SDI_SW12_TIMEOUT_CARD_REMOVAL = 0x64F7,
SDI_SW12_INTERCHAR_PIN_ENTRY = 0x64F8,
SDI_SW12_COMMAND_NOT_ALLOWED = 0x64F9,

SDI_SW12_MAIN_CONNECTION_USED = 0x64FA,
SDI_SW12_INVALID_FILE_CONTENT = 0x64FB,
SDI_SW12_FILE_ACCESS_ERROR = 0x64FC,
SDI_SW12_LOGIC_ERROR = 0x64FD,

SDI_SW12_SDI_PARAMETER_ERROR = 0x64FE,
SDI_SW12_LUHN_CHECK_FAILED = 0x64FF,
SDI_SW12_EXECUTION_ABORTED = 0x6500,
SDI_SW12_EXECUTION_TIMEOUT = 0x6600,

SDI_SW12_MESSAGE_LENGTH_ERROR = 0x6700,
SDI_SW12_NO_SDI_PLUGIN_AVAILABLE = 0x6800,
SDI_SW12_UNKNOWN_PLUGIN_ID = 0x6801,
SDI_SW12_UNKNOWN_PLUGING_ID = 0x6801,

SDI_SW12_INVALID_PLUGIN_RESPONSE = 0x6802,
SDI_SW12_EPP_CONNECTION_ERROR = 0x6900,
SDI_SW12_UNKNOWN_INS_BYTE = 0x6D00,
SDI_SW12_UNKNOWN_CLA_BYTE = 0x6E00,

enum

SDI_SW12_CMAC_ERROR = 0x6FB0,
SDI_SW12_CMAC_LENGTH_ERROR = 0x6FB1,
SDI_SW12_CMAC_MISSING_ERROR = 0x6FB2,
SDI_SW12_ENCRYPTION_ERROR = 0x6FB4,

SDI_SW12_ENCRYPTION_LENGTH_ERROR = 0x6FB5,
SDI_SW12_ENCRYPTION_MISSING_ERROR = 0x6FB6,
SDI_SW12_DECRYPTION_ERROR = 0x6FB8,

SDICLIENT_ERROR {

SDICLIENT_ERROR_NONE = 0,
SDICLIENT_ERROR_COMMUNICATION = -1,
SDICLIENT_ERROR_CONCURRENT_USE = -2,
SDICLIENT_ERROR_CONNECT = -3,

SDICLIENT_ERROR_OVERFLOW = -4,
enum SDICLIENT_ERROR_PARAM = -5,
SDICLIENT_ERROR_OTHER = -6,
SDICLIENT_ERROR_NO_RECEIVE = -7,

SDICLIENT_ERROR_NOT_SUPPORTED = -10,
SDICLIENT_ERROR_NOT_ALLOWED = -11

}

SYSUploadType {

SYS_UPLOAD_SOFTWARE_UPDATE,
SYS_UPLOAD_CONFIG_WHITELIST,
SYS_UPLOAD_CONFIG_SENSITIVE_TAGS,
SYS_UPLOAD_CONFIG_CARD_RANGES,
enum

SYS_UPLOAD_INSTALL_CP_PACKAGE = 11,
SYS_UPLOAD_EMV_CONFIGURATION

}

Functions

enum SDICLIENT_ERROR	getNfcClientError ()
enum SDI_SW12	getNfcSW12 ()

Data Structure Documentation

? libsdi::SdiBase::PluginResult

struct libsdi::SdiBase::PluginResult

Data Fields

int32_t	pluginId	return value of a plugin's `moduleID` function also used as INS byte of SDI 26-xx command to invoke a plugin, 0 if not available
int32_t	responseCode	plugin's `processTrigger` function return value, SDI_SW12_NONE if not available
vector< unsigned char >	responseData	plugin response data

? libsdi::SdiCrypt::Placeholder

struct libsdi::SdiCrypt::Placeholder

Data descriptor for getEncData(), getEncMsgData() and getMsgSignature()

Data Fields

vector< unsigned char >	applicationData	data that can be referenced in Placeholder::tagList (DFA120)
vector< unsigned char >	dataOptions	data formatting options (DFA121) - see SDI programmers guide, getEncData (29-00)
vector< unsigned char >	tagList	DOL format with length 0 for variable lengths (DF8F30)

Macro Definition Documentation

? MSR_CLIENT_ERROR_OFFSET

#define MSR_CLIENT_ERROR_OFFSET 100

if a client error happens, msr functions will return (enum SDICLIENT_ERROR - MSR_CLIENT_ERROR_OFFSET)

? VALIDATION_CHECK_OPTION_RETURN_ALL_MATCHING_RANGES

#define VALIDATION_CHECK_OPTION_RETURN_ALL_MATCHING_RANGES 0x01